Robust Feature Matching and Fast GMS Solution

Singapore University of Technology and Design (SUTD)
Advanced Digital Sciences Center (ADSC)

Speaker: JiaWang Bian (边佳旺)

http://jwbian.net/

June 14, 2017
Content

• Feature Matching Introduction
  • Feature Matching
  • Feature Detector & Descriptor
  • Matching
  • RANSAC-based Geometry Estimation (or Verification)

• Recent Robust Matchers
  • CODE (PAMI,2016)
  • RepMatch (ECCV,2016)

• Fast and Robust GMS Solution(CVPR,2017)
  • Video Demo
  • Methodology
  • Algorithm
  • Share (Material Links)
Feature Matching Introduction
Feature Matching Introduction

• Feature Matching

• Pipeline

Detection → Description → Matching → Geometry
Feature Matching Introduction

• Applications

Correct Correspondences

Geometry between 2 views

Estimate Camera Pose
Localization (SFM)
Tracking (SLAM)
...

Similarity(Number of matches)

Image retrieval
Object Recognition
Loop Closing (SLAM)
Re-localization (SLAM)
...

Applications

Correct Correspondences

Geometry between 2 views

Estimate Camera Pose
Localization (SFM)
Tracking (SLAM)
...

Similarity(Number of matches)

Image retrieval
Object Recognition
Loop Closing (SLAM)
Re-localization (SLAM)
...
Sparse Feature Matching

- Feature detector & descriptor

![Diagram of feature detectors and descriptors with SIFT, SURF, ORB, AKAZE, PCA-SIFT, ASIFT, LIFT, etc.](image)
Feature Matching Introduction

• Matching

Matching

- Nearest-Neighbor
  - Brute-Force
  - Approximate (FLANN)

- Optimization

- Others

Matching Algorithms

- CODE, RepMatch, GMS...
Feature Matching Introduction

• RANSAC-based Geometry Estimation (or Verification)
  • An example for RANSAC framework (fitting a line)

A data set with many outliers for which a line has to be fitted.

Fitted line with RANSAC; outliers have no influence on the result.
Feature Matching Introduction

- **RANSAC-based Geometry Estimation (or Verification)**
  - Fundamental Matrix (for 3D scenes)
    - Point to Line (weak, general)
  - Homography (for 2D scenes)
    - Point to Point (strong, narrow range)
Recent Robust Matchers
Recent Robust Matchers

• CODE[1]
  • For wide-baseline matching.

• RepMatch[2]
  • Based on CODE[1].
  • Solve the repeated structure problem.

---

[1] CODE: Coherence Based Decision Boundaries for Feature Correspondence, IEEE TPAMI, 2016, Lin et. al.
Recent Robust Matchers (CODE)

• Wide-baseline matching
Recent Robust Matchers (CODE)

• Idea

---

**Selected matches**

$(\tau = 0.86)$

- Build regression functions
- Matching likelihood regression
  - Eqn. (20)

**Filtered matches**

- Affine motion regression
  - Eqn. (24), (26)

---

**All matches**

$(\tau = 1.0)$

- Apply cascaded filters
- Likelihood boundaries
  - Eqn. (21)

- Affine boundaries
  - Eqn. (27)

---

Output matches
Recent Robust Matchers (CODE)

- Regression models

![Regression Image]

Fig. 2: Regression can be understood as finding a continuous surface that explains scattered data points (denoted by “+”).

- Likelihood Regression
- Affine motion regression -> x
- Affine motion regression -> y
Recent Robust Matchers (CODE)

- Likelihood Regression
  - Train Data
    - Selected distinctive correspondences (after ratio-test).
  - Test Data
    - All feature correspondences.
  - Features of a correspondence
    - $X_i = [x, y, dx, dy, T_1, T_2, T_3, T_4]$.  
    - $T$ is a transformation matrix of $[s_1, r_1]$ to $[s_2, r_2]$.
    - $s$ means scale, $r$ represents rotation.
  - Labels
    - 1 for all correspondences
  - Cost function
    - Huber function
  - Non-linear Optimization
    - Construct Gaussian Similar Matrix
    - $X$ (Matrix with $n \times n$ elements), $Y$ (Matrix with $nx1$ elements)
    - $n$ is the number of train data
Recent Robust Matchers (CODE)

• Affine motion regression
  • Train Data
    • The inliers of train data in the likelihood model
  • Test Data
    • Correspondences filtered by the likelihood model
  • Feature Space
    • Same as the likelihood model
  • Label
    • $x_2$, and $y_2$. ($x, y$ represents pixel position, 2 means the second image)
  • Cost function
    • Huber function
  • Non-linear Optimization
    • Same as before (Gaussian Similar Matrix).
Recent Robust Matchers (CODE)

• Insight (likelihood model)

![Diagram of Coherent and Incoherent Motion]

Fig. 3: Coherence based separation of true and false matches. Motions are considered coherent if (a) many local points make similar motions or (b) there is broad spatial support for the motion. This is enforced via the likelihood function in Eqn. (21). In contrast, feature matches in (c) and (d) do not give coherent motions, as the matches are not consistent in (c), while there are insufficient smoothly moving points to justify a long-range motion coherence model in (d).
Recent Robust Matchers (CODE)

- Matching samples

- Image pair
- A-SIFT w/o CODE
- CODE input
- A-SIFT w/ CODE
- Alternative display

A

B

C
Recent Robust Matchers (CODE)

- Structure from Motion

[Image: A set of multi-view images [43]]

[Image: Agisoft [48]: A commercial 3D reconstruction software]

[Image: Visual SfM [3], [44], [45], [46], [47]]

[Image: Visual SfM using feature matches returned by A-SIFT w CODE]

Recent Robust Matchers (CODE)

• Run time comparison
Recent Robust Matchers (RepMatch)

• RepMatch

Input Images

(a) Visual SfM

(b) Visual SfM with our matches

(c) Dense reconstruction
Recent Robust Matchers (RepMatch)

- Repetitive Structure

(a) *All matches*  (b) *Epipolar*  (c) *BF*  (d) *RepMatch*

Illustration on real images. Black dots in (a) & (b) indicate wrong matches. Note: Common central tower belong to physically different parts of the building.
Recent Robust Matchers (RepMatch)

• Idea

Core-set and Local hypothesis

Geometrically Consistent

Geometrically Inconsistent

Local match consistency

Inlier Set

BF training and classification

Final Matches

• : inlier
  ○ : outlier (removed)
Recent Robust Matchers (RepMatch)

- Structure from Motion
Recent Robust Matchers (RepMatch)

- Structure from Motion
Recent Robust Matchers (RepMatch)

• Structure from Motion

(a) Visual SfM

(b) Visual SfM with $BF$ matches

(c) Visual SfM with $RepMatch$

(ii) City street scene
Recent Robust Matchers (RepMatch)

- Structure from Motion

(d) \textit{RepMatch} based reconstruction

(e) \textit{RepMatch} based normal map

(iii) Building scene
Fast and Robust GMS Solution
Video Demo

• ORB with GMS vs SIFT with Ratio
Motivation: Trade-off of quality and speed

- Trade-off

Matching

Nearest-Neighbor

Ratio test
  Popular, Fast, Non-Robust

GMS
  Fast, Robust

Current Methods

Graph Matching

Optimization

Slow, Robust
Methodology: Motion Smoothness

• Observation
  • True matches (green) are visually smooth while false matches (cyan) are not.
Methodology: Key idea

• Inference
  • According to the Bayesian rule, as true matches are smooth in motion space, consistent matches are thus more likely to be true.

• Key idea
  • Find smooth matches from noisy data as our proposals.

• Method
Methodology: Motion Statistics

- Motion Statistics Model

\[ S_i = |x_i| - 1, \]
Methodology: Motion Statistics

• Distribution

\[ S_i \sim \begin{cases} 
   B(n, p_t), & \text{if } x_i \text{ is true} \\
   B(n, p_f), & \text{if } x_i \text{ is false}
\end{cases} \]

• Let \( f_a \) be one of the \( n \) supporting features in region \( a \)

• Let \( p_t, p_f \) be the probability that, feature \( f_a \)'s nearest neighbor is in region \( b \), given \( \{a, b\} \) view the same and different location, respectively,
Methodology: Motion Statistics

• Event

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_a^t$</td>
<td>$f_a$ matches correctly, $p(f_a^t) = t$</td>
</tr>
<tr>
<td>$f_a^f$</td>
<td>$f_a$ matches wrongly, $p(f_a^f) = 1 - t$</td>
</tr>
<tr>
<td>$f_a^b$</td>
<td>$f_a$’s nearest-neighbor is a feature in region $b$</td>
</tr>
</tbody>
</table>

• Assumption

$$p(f_a^b | f_a^f) = \beta m / M$$

Here, $m$ is the number of features in region $b$ and $M$ is the number of features in second image. $\beta$ is a factor added to accommodate violations of assumption caused by repeated patterns.
Methodology: Motion Statistics

- Probability

\[ p_t = p(f_a^t) + p(f_a^f) p(f_a^b | f_a^f) \]
\[ = t + (1 - t) \beta \frac{m}{M} \]

Explanation: If \( \{a, b\} \) view the same location, event \( f_a^b \) occurs when \( f_a \) matches correctly or it matches wrongly but coincidentally lands in region \( b \).

\[ p_f = p(f_a^f) p(f_a^b | f_a^f) \]
\[ = \beta (1 - t) \left( \frac{m}{M} \right) \]

Explanation: If \( \{a, b\} \) view the different location, event \( f_a^b \) occurs only when \( f_a \) matches wrongly and coincidentally lands in region \( b \).
Methodology: Motion Statistics

• Multi-region Generalization

\[ S_i = \sum_{k=1}^{K} | \mathcal{X}_{a^k b^k} | - 1 \]
Methodology: Motion Statistics

- Distribution

\[ S_i \sim \begin{cases} 
B(Kn, p_t), & \text{if } x_i \text{ is true} \\
B(Kn, p_f), & \text{if } x_i \text{ is false}
\end{cases} \]

- Mean & Variance

\[
\begin{align*}
\{m_t = Kn p_t, s_t = \sqrt{Kn t (1 - p_t)}\} & \quad \text{if } x_i \text{ is true} \\
\{m_f = Kn p_f, s_f = \sqrt{Kn p_f (1 - p_f)}\} & \quad \text{if } x_i \text{ is false}
\end{align*}
\]
Methodology: Motion Statistics

• Analysis
  • Partitionability

\[ P = \frac{m_t - m_f}{s_t + s_f} = \frac{Knp_t - Knp_f}{\sqrt{Knp_t(1 - p_t)} + \sqrt{Knp_f(1 - p_f)}} \]

• Quantity-Quality equivalence:

\[ P \propto \sqrt{Kn}. \]

• Relationship to Descriptors:

\[ \lim_{t \to 1} m_t \to Kn, \quad \lim_{t \to 1} m_f \to 0, \quad \lim_{t \to 1} P \to \infty. \]
Methodology: Motion Statistics

• Experiments on real data:

The model is evaluated on Oxford Affine Dataset. Here, we run SIFT matching and label all matches as inlier or outlier according to the ground truth. We count the supporting score for each match in a small region.
Algorithm: Grid Framework

• Grid Framework
  • Both images are segmented by a pre-defined grid.
  • Calculating the Motion Statistics for cell-pairs instead of each feature correspondence.

O(N) $\rightarrow$ O(1)!
Algorithm: Motion Kernels

- Basic Motion Kernel

\[ S_{ij} = \sum_{k=1}^{K=9} |\chi_{ik}^{j} j^{k}| \]
Algorithm: Motion Kernels

- Generalized Motion Kernels (Extension*)
  - Rotation
    - Fixed
      - **\(a^1 a^2 a^3\)**
      - **\(a^4 a a^6\)**
      - **\(a^7 a^8 a^9\)**
    - (1)
      - **\(b^1 b^2 b^3\)**
      - **\(b^4 b b^6\)**
      - **\(b^7 b^8 b^9\)**
    - (2)
      - **\(b^4 b^1 b^2\)**
      - **\(b^7 b b^3\)**
      - **\(b^8 b^9 b^6\)**
    - (3)
      - **\(b^7 b^4 b^1\)**
      - **\(b^8 b b^2\)**
      - **\(b^9 b^6 b^3\)**
    - (4)
      - **\(\ldots\)**
  - Scale
    - Varying the cell size of the second image by a scale factor.
Algorithm: Empirical parameters

• How many grid-cells should be used?
  • Too fine: weak statistics and low efficiency.
  • Too coarse: low accuracy
  • The empirical results show 20 x 20 is a good choice.

• How to set the threshold?

\[
\tau = m_f + \alpha s_f \quad \tau \approx \alpha s_f \approx \alpha \sqrt{n}
\]

cell-pair \( \{i, j\} \) \( \in \) \[
\begin{cases} 
\mathcal{T}, & \text{if } S_{ij} > \tau_i = \alpha \sqrt{n_i} \\
\mathcal{F}, & \text{otherwise}
\end{cases}
\]
Algorithm: GMS

- Grid Motion Statistics Algorithm

**Algorithm 1 Grid Motion Statistics**

\[ \text{Input: } X', s, r \{\text{Correspondences, scale, rotation}\} \]

\[ \text{Output: } \text{Inliers} \]

\[ G_1, G_2 = \text{GenerateGrids}(s) \]
\[ K = \text{GenerateMotionKernel}(r) \]

for \( i = 1 \) to \( |G_1| \) do

\[ j = 1; \]

for \( k = 1 \) to \( |G_2| \) do

if \( |X_{ik}| > |X_{ij}| \) then

\[ j = k; \]

end if

end for

\[ S_{ij}, \tau_i = \text{ComputeGMS}(K) \{\text{Eq. (13)(14)}\} \]

if \( S_{ij} > \tau_i \) then

\[ \text{Inliers} = \text{Inliers} \cup X_{ij}; \]

end if

end for

Repeat algorithm with grid patterns shifted by half cell-width in the \( x, y \) and both \( x \) and \( y \) directions.

\[ \text{return } \text{Inliers} \]
Algorithm: Full Feature Matching

- Full feature matching pipeline

---

**Algorithm 2 Feature Matching with GMS**

**Input:** \( I_a, I_b, \text{Scale}, \text{Rotation} \) \{Two input images\}

**Output:** \( \text{Inliers} \)

Extract Features and Descriptions: \( F_a, D_a, F_b, D_b \)

Find Nearest Neighbour Matches: \( \mathcal{X} \)

Initialise \( \text{Inliers} \) and \( \text{number} \)

\( \text{number} = 0 \)

for all \( s \in \text{Scale} \) do

  for all \( r \in \text{Rotation} \) do

    \( \text{inlier} = gms(\mathcal{X}, s, r) \)

    if \( |\text{inlier}| > \text{number} \) then

      \( \text{number} = |\text{inlier}| \)

      \( \text{Inliers} = \text{inlier} \)

    end if

  end for

end for

return \( \text{Inliers} \)
Algorithm: Run time

• Run time on Image pairs
  • ORB feature extraction (about 35ms on cpu)
  • Nearest Neighbor Matching (106ms on cpu, 25ms on gpu)
  • GMS (1ms on cpu)
  • Overall: \( \frac{1000}{(2 \times 35 + 25 + 1)} = 10.42 \text{fps} \)

• Real time on Video data
  • ORB and NN can run parallelly on video sequence.
  • Overall: \( \frac{1000}{35} = 28.57 \text{fps} \)
Evaluation

• Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TUM [38]</th>
<th>Strecha [37]</th>
<th>VGG [25]</th>
<th>Cabinet [38]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full name</td>
<td>RGB-D SLAM Dataset and Benchmark</td>
<td>Dense Multiview Stereo Dataset</td>
<td>Affine Covariant Regions Datasets</td>
<td>A subset of TUM dataset</td>
</tr>
<tr>
<td>Image pairs</td>
<td>3141</td>
<td>500</td>
<td>40</td>
<td>578</td>
</tr>
<tr>
<td>Ground truth</td>
<td>Camera pose, Depth</td>
<td>Camera pose, 3D model</td>
<td>Homography</td>
<td>Camera pose, Depth</td>
</tr>
<tr>
<td>Description</td>
<td>Including all image condition changes</td>
<td>Well-textured images</td>
<td>Viewpoint change, zoom+rotation, blur</td>
<td>Low-texture images with strong light</td>
</tr>
</tbody>
</table>

• Capture of TUM dataset
Evaluation

• Capture of Strecha dataset

• Capture of VGG dataset
Evaluation

• Matching ability

![Graphs showing recall, precision, and F-measure for TUM and Strecha datasets with different image rotation degrees and image pair order for VGG models.](image-url)
Evaluation

- Pose Estimation
Evaluation

- Wide-baseline matching

In both graphs, the first row shows initial results and the second row illustrates the matches after RANSAC.
Evaluation

• GMS on Images with Repetitive Structures

Images are selected by [1], where many state-of-art matchers fail and SIFT fails all.

Evaluation

• Non-rigid object
Evaluation

• Video Demo(screen shot)
Share

- JiaWang’s Home Page
  - http://jwbian.net/
- Project Page
  - http://jwbian.net/gms/
- Code on GitHub:
  - https://github.com/JiawangBian/GMS-Feature-Matcher
- Videos on YouTube:
  - https://youtu.be/3SlBqspLbxI
- Links to CODE and RepMatch
  - http://www.kind-of-works.com/
Q&A